



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/035,800	12/28/2001	Sudhindra P. Herle	2002.02.002.WTO	2037
Docket Clerk 7590 06/30/2010				
P.O. Drawer 800889				
Dallas, TX 75380				
EXAMINER				
PITARO, RYAN F				
ART UNIT		PAPER NUMBER		
2174				
MAIL DATE		DELIVERY MODE		
06/30/2010		PAPER		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES

---

*Ex parte* SUDHINDRA P. HERLE  
and MARK MITCHELL

---

Appeal 2009-005256  
Application 10/035,800  
Technology Center 2100

---

Decided: June 30, 2010

---

Before JAMES D. THOMAS, HOWARD B. BLANKENSHIP, and  
JAMES R. HUGHES, *Administrative Patent Judges*.

BLANKENSHIP, *Administrative Patent Judge*.

DECISION ON APPEAL

STATEMENT OF THE CASE

This is an appeal under 35 U.S.C. § 134(a) from the Examiner's final rejection of claims 1, 2, 5-9, 12-15, and 17-22, which are all of the remaining claims in the application. We have jurisdiction under 35 U.S.C. § 6(b).

We affirm.

*Invention*

Appellants' invention relates to a wireless communication device that stores a system GUI configuration file and a downloaded service provider GUI configuration file, each of which contains a text name checksum value that is calculated using the text name of the respective GUI parameter data. After the download operation is complete, a text name checksum comparator program compares the downloaded text name checksum value to the initial text name checksum value in the system GUI configuration file. If the two text name checksum values do not match, the downloaded service provider GUI configuration file is rejected. Abstract.

*Representative Claim*

1. A wireless communication device comprising:  
a main controller capable of executing a basic operating system application program that operates communication functions of said wireless communication device and that controls a first graphical user interface (GUI) for interacting with a user;  
a memory, within the wireless communication device, coupled to said main controller, capable of storing a first GUI configuration file and a second GUI configuration file; wherein  
said first GUI configuration file contains first GUI parameter data comprising  
a first plurality of text names and,

a corresponding plurality of data comprising at least one of:  
sounds, graphical images, text, menu options and a menu hierarchy  
associated with said first graphical user interface, and  
a first text name checksum value calculated from only said first  
plurality of text names, and  
said second GUI configuration file contains second GUI  
parameter data comprising  
a second plurality of text names and,  
a corresponding plurality of data comprising at least one of:  
sounds, graphical images, text, menu options and a menu hierarchy  
associated with a second graphical user interface, and  
a second text name checksum value calculated from only said  
second plurality of text names; and  
wherein said main controller is operable to validate said second  
GUI parameter data by comparing said first text name checksum value  
contained in said first GUI configuration file with said second text  
name checksum value contained in said second GUI configuration  
file.

*Examiner's Rejections*

Claims 1, 2, 5-9, 12-15, and 17-22 stand rejected under 35 U.S.C.  
§ 103(a) as being unpatentable over Martin (US 6,509,913 B2), Brodersen  
(US 6,324,693 B1), and Bodnar (US 6,544,295 B1).

### *Claim Groupings*

In view of Appellants' arguments in the Appeal Brief, we will decide the appeal on the basis of the claims we address *infra*. See 37 C.F.R. § 41.37(c)(1)(vii).

### PRINCIPAL ISSUE

Have Appellants shown that the Examiner erred in finding that the combination of Martin, Brodersen, and Bodnar teaches a “first text name checksum value calculated from only said first plurality of text names” as recited in claim 1?

### FINDINGS OF FACT

#### *Martin*

1. Martin teaches techniques for configuring user interfaces (e.g., man-machine interfaces) for wireless devices. A network operator can configure (replace, alter, or customize) user interfaces. The configuring or customization also enables network operators to provide options, logos, and advertising in a controllable way. Abstract.

2. Wireless terminals or devices, such as cellular telephones, pagers and Personal Digital Assistants (PDAs), contain interfaces between the user and the machine. Such interfaces are referred to as user interfaces or man-machine interfaces (MMIs). These interfaces determine how a user is able to interact with the terminal or device. Typically, the user interfaces include a display device (e.g., a LCD display) which displays information or choices for the user of the terminals or devices, and the user navigates through the information or choices with buttons. Col. 1, ll. 19-28. A major

disadvantage of fixed MMIs available with conventional mobile telephones is that the MMIs are not able to be modified following manufacture of the mobile telephones. Col. 1, ll. 46-48.

3. With a configurable man-machine interface, the user interface can be modified or configured after the mobile telephone is manufactured. Title; col. 3, ll. 29-34. The screen configuration information provided to the remote wireless browser 220 is a configuration file that directs the remote wireless browser 220 in arranging the screen displayed on the display 218. The configuration file is downloaded by the user interface controller 212 within the network gateway 208 to the remote computing device 216. Col. 5, ll. 31-37; col. 11, ll. 1-14.

4. Figure 2B is a diagram of a representative configured screen 250 that is displayed on the display 218. The configured screen is suitable for use when the remote computing device is a mobile phone. The representative configured screen 250 includes a plurality of components C1-C8 that together form the configured screen 250. Each of these components C1-C8 are determined and arranged by the screen configuration information, and the contents of the screens can also be controlled by the screen configuration information. For example, with respect to the representative configured screen 250, the components C1-C8 can be assigned locations on the screen as well as contents to be displayed within each of the components. The screen configuration information is provided by a markup or script language or other hypermedia that provides a description of the desired screen (MMI). The contents for the components can be a menu list, a button, or an image (e.g., advertisement or logo). For example, the screen

configuration information could assign the contents for each of the components as indicated in Table 1.

TABLE 1

<u>MMI Component</u>	<u>Content</u>
C1	http://operator.com/ad-101
C2	internal:dialing-screen
C3	http://operator.com/logo
C4	http://operator.com/lookup
C5	internal:newsmenu
C6	internal:weathermenu
C7	internal:clearbutton
C8	internal:redialbutton

In this example, the screen is configured into eight (8) components (MMI components). Each of the components is assigned different contents that are to be depicted on the screen in the corresponding location of the component on the screen. The contents for the components are identified by a resource locator, such as hypermedia link name (e.g., a Universal Resource Locator (URL)) or hypermedia function name. Col. 6, ll. 5-53.

5. Each of the components associated with a screen to be displayed can have an associated URL. Default MMI components can be used, such as a default URL stored in local memory 224 in the remote computing device 216. The default MMI components have a URL that begins with “internal” to indicate the associated MMI component belongs to the set of default MMI components. Other MMI components can use external resources that are identified by URLs designating remote locations. Col. 7, ll. 20-30.

6. A configuration file can include screen configuration information that is used to update an alias table. The alias table can contain a single entry for each MMI component. Each entry in the alias table indicates the appropriate URL for obtaining the content of the component. The alias table is stored in the local memory 224 of the remote wireless computing device 216 which is connected to and communicates with the remote wireless browser 220. Table 2 below illustrates a representative alias table. The alias table associates an alias component name with a URL for the content for the component.

TABLE 2

<u>MMI Component</u>	<u>Content URL</u>
top-menu	http://operator.com/menu.wml
dialing-screen	internal:dialing-screen

The alias “top-menu” is a MMI component that is used to look-up the actual URL “http://operator.com/menu.wml” in the alias table. Similarly, the alias “dialing-screen” corresponds or maps to the actual URL “internal:dialing screen” where “internal” is indicative of a default MMI component. The alias table allows the MMI components for the remote wireless browser to be relocated or changed without having to reprogram or physically alter the remote wireless browser’s operation. The MMI can be easily customized or provided with any user services as deemed suitable. Col. 8, ll. 33-53; col. 11, ll. 9-14.

*Brodersen*

7. Brodersen teaches informing a remote user that the central database has been updated, and that the user’s docking operation is limited to downloading changes to the database; that is, the user is not permitted to



upload transactions until his own partially replicated database has been upgraded to match the central database schema. The user then downloads the database upgrade files corresponding to the upgrade installed on the central database. The installation component is invoked (either by the client-side software upon detecting the upgrade condition, or expressly invoked by the user, depending on the upgrade file's attributes). The installation component opens the database upgrade files and contains a checksum value for the upgrade file set. The checksum is compared against a value in the user's configuration file that uniquely identifies the user's software level. If the checksum value is different, then the client software does not have the ability to apply the schema changes. The schema changes will be deferred until the user separately upgrades his or her software level. Col. 17, ll. 42-60.

8. A checksum may also be included in the DX files that represent transactions. The data merge component verifies the checksum associated with the transaction against the checksum in the configuration file. If the checksums do not match, the application of the DX file is deferred. This prevents a user from inadvertently applying changes that cannot be supported by the software. Col. 17, ll. 61-67.

*Bodnar*

9. Bodnar teaches a system that notifies a user when content for a "mark," or Web site, has changed. Col. 18, l. 65 to col. 19, l. 7.

10. A checksum is used to detect a change in content. However, applying a checksum in and of itself is not a particularly useful indicator of content change for HTML-based forms. Of interest is not whether the HTML content changed, but whether it changed in an interesting way (i.e.,

in a way relevant to the user). Items which are not of interest include items that are constantly cycling, such as advertisements or bulletins which continually change. A checksum approach, if it did not take into account this changing uninteresting material, would not fare much better than the HTTP-header checking approach, since the content being checked would always be identified as new. Col. 19, ll. 25-40.

11. An HTML document or form comprises two components: text and markup (HTML formatting codes). A “selective checksum” is employed. Specifically, a checksum is derived from the interesting text, while ignoring markups, thereby yielding a reliable indicator of whether information of interest to the user has changed. Col. 19, ll. 41-48.

12. A checksum can be constructed by adding all the units together which comprise the content of interest, such as adding together all of the byte values for the characters of the text portion of the body of an HTML document. Col. 19, ll. 49-54.

13. The method initializes or resets the checksum and time stamp (e.g., by setting each equal to the value of zero). A do/while loop is established for hopping or jumping from HTML tag to HTML tag, paying particular attention to any TITLE tag which is encountered. The top of an HTML document typically comprises a header section followed by a body. Generally, the method is less interested in text in the header, as that often includes changing text which is not really of interest to the user. What really is of interest to the user is change to content in the body of the HTML form. The approach adopted, therefore, is to hop from tag to tag until a region of interest (i.e., BODY tag) is encountered. During this hopping process, if a tag is encountered representing a TITLE, the method copies the text of the

title, for appending it to the bulletin. Upon completion of the do/while loop, the method will have arrived at the body of the HTML (if any). Col. 25, ll. 24-42.

## PRINCIPLES OF LAW

### *Claim Interpretation*

During examination, claims are to be given their broadest reasonable interpretation consistent with the specification, and the language should be read in light of the specification as it would be interpreted by one of ordinary skill in the art. *In re Amer. Acad. of Sci. Tech Ctr.*, 367 F.3d 1359, 1364 (Fed. Cir. 2004) (citations omitted). The Office must apply the broadest reasonable meaning to the claim language, taking into account any definitions presented in the specification. *Id.* (citations omitted).

### *Obviousness*

“What matters is the objective reach of the claim. If the claim extends to what is obvious, it is invalid under § 103.” *KSR Int’l Co. v. Teleflex, Inc.*, 550 U.S. 398, 419 (2007). “The combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results.” *Id.* at 416.

## ANALYSIS

### *Section 103 rejection of claims 1, 6-8, and 15*

Appellants contend that Martin does not teach a first plurality of text names and a corresponding plurality of data as recited in claim 1. App. Br. 14. In particular, Appellants contend that identifying components in a

screen configuration file using an ordinal identifier such as C1 does not teach a first plurality of text names. Appellants also contend that using alias names in a memory structure updated from a GUI configuration file does not teach a GUI configuration file containing a plurality of text names. App. Br. 15.

We broadly but reasonably construe Appellants' recited "text names" to simply mean text identifiers (e.g., file names), and the disputed limitations of Appellants' claim 1 directed to "text names" to simply require some alphabetical characters identifying data. *See In re Am. Acad. of Sci. Tech Ctr.*, 367 F.3d at 1364. Martin describes a memory that contains configuration information including GUI configuration file data and alias names. FF 6. We understand Martin's alias names to be alphabetical characters identifying data. Appellants have not provided evidence or persuasive argument to distinguish "a memory ... capable of storing a first GUI configuration file [that] contains a first plurality of text names" from a memory that contains configuration information including GUI configuration file data and alias names as taught by Martin.

The Examiner relies on Bodnar to teach calculating a checksum from a text portion in a region of interest of an HTML document. Ans. 7-8. Appellants contend that Bodnar describes calculating a checksum that jumps from tag to tag in an HTML file, summing only the content text and not the tag text. Thus, according to Appellants, Bodnar describes calculating a checksum from only content text data, not from only the tag text identifiers. App. Br. 15-16.

Appellants appear to base their argument on the premise that the scope of the term "text names" as recited in claim 1 is limited to the tag text

identifiers taught by Bodnar. The Examiner does not find that scope of the “text names” is so limited, and we agree with the Examiner. As we explain *supra*, Martin teaches “text names.” The Examiner finds that calculating a checksum from a “text name” i.e., from text, encompasses calculating a checksum from text in a region of interest of the HTML document as taught by Bodnar. Ans. 8. The record supports the Examiner’s finding. FF 11-13. Appellants have not provided evidence or persuasive argument to distinguish calculating a checksum from “text names” from Bodnar’s teaching of calculating a checksum from text in a region of interest of a document.

Appellants contend that in claim 1, text names are identifiers, and that claim 1 operates to detect a change in the identifiers of content in a GUI configuration file. Reply Br. 1-2. However, claim 1 does not recite “text names are identifiers” and “detect a change in the identifiers of content in a GUI configuration file,” and Appellants have provided no basis for reading these limitations into claim 1. Even so, as we explain *supra*, the combination of Martin and Bodnar teaches calculating a checksum from text identifiers.

Appellants also repeat their contention that Bodnar describes checksumming content of a web page, not the tags that identify the content. Reply Br. 2. Appellants’ argument does not address the Examiner’s finding that Bodnar teaches calculating a checksum value from text in a region of interest, which is equivalent to calculating a checksum from a “text name,” within the meaning of claim 1.

The Examiner finds that Martin teaches replacing a GUI configuration file on a device with a downloaded GUI configuration file. Ans. 3-4. The Examiner finds that Brodersen teaches a checksum value to indicate whether

the downloaded file is suitable for processing by the device software. Ans.

4. The Examiner finds that the checksum value can be calculated from text names as taught by Bodnar. Ans. 4-5, 8. The record supports the Examiner's finding. FF 1-13. Appellants' arguments do not show error in the Examiner's findings.

Appellants do not present arguments for the separate patentability of claims 6-8 and 15, but instead rely on the arguments presented for claim 1, which we find unpersuasive.

*Section 103 rejection of claims 2 and 9*

Appellants contend that column 18 of Brodersen does not teach the limitations of claims 2 and 9. App. Br. 16-17, 23-24. However, the Examiner finds that column 17 of Brodersen teaches the limitations of claims 2 and 9. Ans. 8. Appellants have not responded to this finding.

*Section 103 rejection of claims 5 and 12*

Appellants contend that Martin teaches screen configuration information that may include either or both of locally stored default components and components defined by external resources. Appellants further contend that Martin does not teach a default file of screen configuration information. App. Br. 18, 25. The Examiner finds that the GUI configuration file of Martin would first rely on all default internal files and would in fact be a default configuration file if no components were updated. Ans. 8. The record supports the finding. FF 2, 5. Appellants have not addressed the Examiner's finding.

*Section 103 rejection of claim 13*

Appellants contend that nothing in Martin, Brodersen, or Bodnar teaches a cellular telephone handset in the context of (intervening) claim 9. App. Br. 26. The Examiner finds that Martin teaches a cellular telephone handset. Ans. 5. The record supports the finding. FF 2. Appellants have not addressed the Examiner's finding.

*Section 103 rejection of claim 14*

Appellants contend that nothing in Martin, Brodersen, or Bodnar teaches a personal digital assistant device in the context of (intervening) claim 9. App. Br. 26. The Examiner finds that Martin teaches a personal digital assistant device. Ans. 5. The finding is supported by the record. FF 2. Appellants have not addressed the Examiner's finding.

*Section 103 rejection of claims 17-20 and 22*

Appellants contend that nothing in Martin, Brodersen, or Bodnar teaches a service provider GUI configuration file. App. Br. 30-33. The Examiner relies on column 5 of Martin to teach the service provider GUI configuration file. Ans. 6. The finding is supported by the record. FF 3. Appellants have not addressed the Examiner's finding.

*Section 103 rejection of claim 21*

Appellants contend that the Examiner has not shown where the combination of Martin, Brodersen, and Bodnar teaches a system default GUI configuration file as recited in claim 21. App. Br. 32. The Examiner finds that the initial GUI configuration file of Martin is a system default GUI

configuration file. Ans. 8. The finding is supported by the record. FF 2, 5. Appellants have not addressed the Examiner's finding.

*Conclusion*

Based on the foregoing, we are not persuaded that any claim has been rejected in error. We sustain the rejections under 35 U.S.C. § 103(a).

CONCLUSION OF LAW

Appellants have not shown that the Examiner erred in finding that the combination of Martin, Brodersen, and Bodnar teaches a "first text name checksum value calculated from only said first plurality of text names" as recited in claim 1.

DECISION

The rejection of claims 1, 2, 5-9, 12-15, and 17-22 under 35 U.S.C. § 103(a) as being unpatentable over Martin, Brodersen, and Bodnar is affirmed.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a). *See* 37 C.F.R. § 41.50(f).

AFFIRMED



Appeal 2009-005256  
Application 10/035,800

msc

Docket Clerk  
P.O. Drawer 800889  
Dallas, TX 75380